NORDA Technical Note 272

Naval Ocean Research and
Development Activity
NSTL, Mississippi 39529

CHORDS: A New Temperature or Sound Speed Profile Thinning Algorithm

Approved for Public Release
Distribution Unlimited

George Kerr
Ocean Acoustics and Technology Directorate
Numerical Modeling Division

May 1984

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NORDA Tecnnical Note 272	2. GOVT ACCESSION NO. AD-A145464	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) CHORDS: A New Temperature or Sound Speed Profile Thinning Algorithm		5. TYPE OF REPORT & PERIOD COVERED Final
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) George Kerr		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Ocean Research & Development Activity Ocean Acoustics and Technology Directorate NSTL, Mississippi 39529		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS P.E. 940101
11. CONTROLLING OFFICE NAME AND ADDRESS Same		12. REPORT DATE
		13. NUMBER OF PAGES 30
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release
Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

profile thinning
sound speed
temperature
digital data
profile filtering

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A new sound velocity and temperature profile thinning algorithm especially designed for use with high vertical resolution (1 meter) conductivity-temperature-depth (CTD) or expendable bathythermograph (XBT) digital data is described. The thinning operation can be halted once the desired number of output points is obtained or alternatively when a user-supplied tolerance value is reached. The new algorithm has the advantage of scanning the entire profile during each iteration. As a result, unlike other thinning algorithms examined, profile curvature at depth is retained.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ABSTRACT

A new sound speed and temperature profile thinning algorithm especially designed for use with high vertical resolution (1 meter) conductivity-temperature-depth (CTD) or expendable bathythermograph (XBT) digital data is described. The thinning operation can be halted once the desired number of output points is obtained or alternatively when a user-supplied tolerance value is reached. The new algorithm has the advantage of scanning the entire profile during each iteration. As a result, unlike other thinning algorithms examined, profile curvature at depth is retained.

ACKNOWLEDGMENTS

This work is a description of a part of the software developed by the author as a part of the Naval Ocean Research and Development Activity's (NORDA's) exercise data handling and acoustic model preparation package NORDAPS. The author wishes to thank Mr. Richard Lauer, NORDA Code 223, for his support and most helpful suggestions during the preparation of this report.

This project was funded by the Tactical Anti-Submarine Warfare Environmental Acoustic Support Project (PE 940101), Mr. Charles Stuart, project manager.

CHORDS: A NEW TEMPERATURE OR SOUND SPEED PROFILE THINNING ALGORITHM

I. INTRODUCTION

Digitally recording oceanographic sensors have the capability of gathering hundreds or even thousands of data points in a single cast. In contrast, present acoustic models are capable of utilizing only a small number (on the order of 30) of these points. It is, therefore, important to have a reliable method of thinning the digital data to the number of points acceptable to the acoustic models. In addition to speed, automated methods offer the advantage of uniformity over "by hand" operations.

Two automated thinning algorithms (NORDA's FRITZ and an algorithm obtained from the SACLANT Center) were examined to determine their performance with T7-XBT profiles with 1 meter (m) resolution from the surface to 100 m and 10 m resolution from 100 m to the bottom of the XBT (approximately 800 m). Both algorithms were similar in function with the SACLANT Center algorithm being somewhat more sophisticated. These algorithms contained a particular flaw that made them unacceptable. Both algorithms started processing at at the surface and worked down the profile. As a result the bottom portion of the temperature profile was often not reached. In other words, the desired number of output points were found before the entire profile was operated on by the algorithms. As a result, the curvature in the bottom portion of the measured profile was removed from the thinned profile.

Figure 1 illustrates a complex staircased sound speed profile of approximately 1 m resolution. Figure 2 illustrates the same sound speed profile thinned to 30 points using the SMOOTH thinning algorithm (similar in methodology to the algorithms mentioned above) incorporated into PRISM, an acoustic analysis model. Notice the change in the sound speed gradient in the layer, the loss of some detail in the staircase near 100 m, and the loss of profile curvature below 400 m. The loss of curvature in this case resulted in the movement of the deep sound channel axis from its actual depth of near 800 m to a depth of approximately 1000 m.

Because of this type of performance, it was decided to design a thinning algorithm that would insure that the entire profile was considered during each point selection iteration of the algorithm. The result is a FORTRAN subroutine named CHORDS.

II. PROGRAM GENERAL DESCRIPTION

As the name implies, CHORDS fits straight line segments (chords) to temperature or sound speed profiles (a version of CHORDS is available for both). The chords are selected so that the maximum difference between any new chord and the profile segment it encompasses is diminished as each chord is added to the chord set defining the entire profile.

The CHORD process begins with the selection of an initial set of significant points. For a temperature profile the initial significant points are the first (surface) and last (bottom) points of the profile, and a point selected to represent the mixed layer depth. The mixed layer depth point is selected as the (n-1)th point where the nth point temperature differs by at least ±.2°C from the surface temperature. The three resulting significant points then define an initial set of two chords.

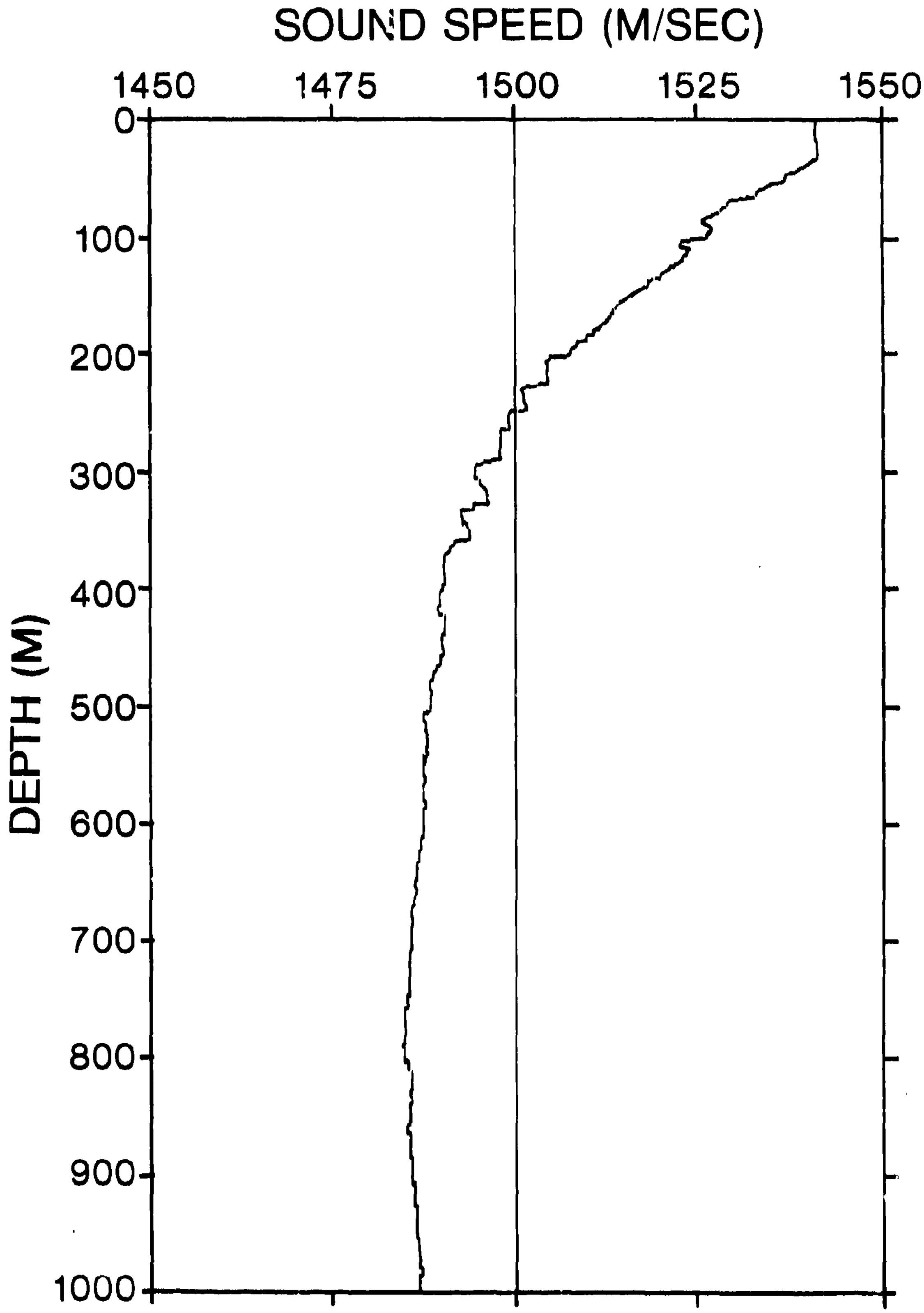

Figure 1. Original, approximately 1-meter resolution sound speed profile

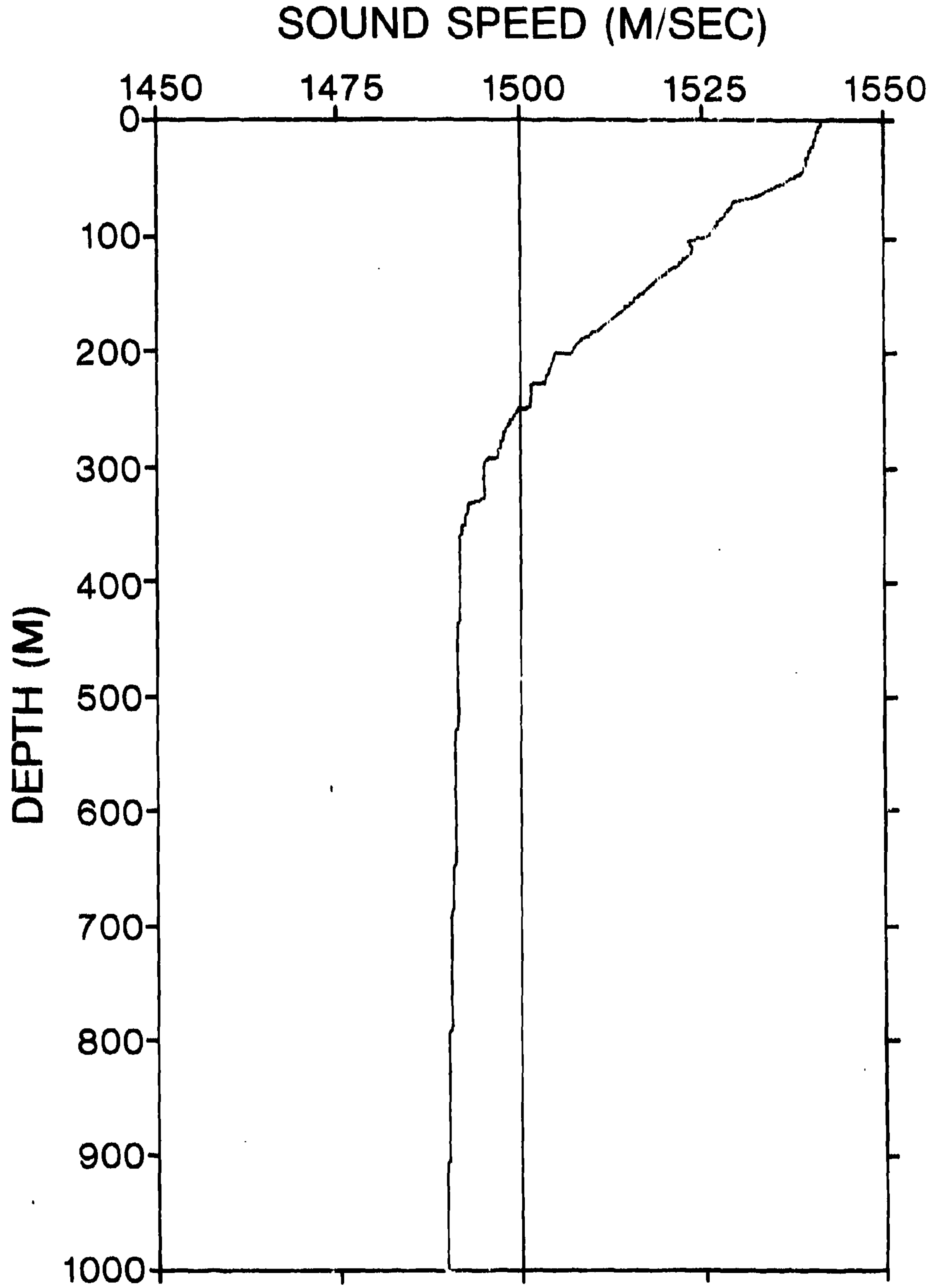

Figure 2. SMOOTH thinned 30-point representation of the original profile (Fig.1)

For a sound speed profile, the significant points are the first and last profile points and points defining the layer depth and depth of the deep sound channel axis. The deep sound channel axis depth is defined as the depth at which the absolute sound speed minimun occurs. The layer depth is defined as the depth above the deep sound channel axis depth where the local maximum sound speed occurs. Thus for sound speed profiles an initial set of up to three chords is possible. Figure 3 illustrates an initial point and chord set.

The algorithm then treats each chord and associated profile segment in the following manner:

A--The chord and corresponding profile segment are rotated and translated (Figure 4) so that the chord becomes the new temperature (sound speed) axis.

B--The absolute maximum of the transformed profile segment point set is then located (Figure 5). This point, defining the maximum absolute difference between the profile segment and its associated chord, becomes a new significant point candidate.

After all chords are processed, the candidate significant points are compared. The candidate point having the largest absolute difference between itself and its associated chord of all such pairs is selected as a new significant point. The new point divides an existing profile segment into two parts (Figure 5), thereby creating two chords in place of a single previously existing chord. By retaining the chord to profile segment maximum absolute differences of all chord-profile segment pairs, further iterations will only require two chord-profile segment transformations and maximum difference calculations (one iteration for each of the two newly created chords).

The above chord-splitting process will continue until the number of desired points is found, or until the maximum allowable absolute difference between any chord and its encompassed profile segment is less than a user-defined tolerance limit. In tests of the algorithm's performance, a metric unit tolerance value of 0.05 for temperature and 0.1 for sound speed were found to adequately reproduce profile shapes and yet reduce the number of profile points by approximately a factor of 10.

III. PROGRAM EXAMPLE

Figure 6 shows the test sound speed profile (Figure 1), addressed previously, thinned to 30 points using the CHORDS algorithm. Notice how the in-layer gradient of the original profile has been retained along with much of the staircase structure. Notice also how the curvature of the profile below 400 m has been retained. Here the shape of the original profile is well-matched in the vicinity of the deep sound channel, and the depth of the deep sound channel axis has been retained.

Appendix A gives a more detailed description of the subroutine in addition to instructions for its use. Appendix B contains a listing of the program. Appendix C contains a description of the subroutines, functions, and variable names used in the program.

IV. CONCLUSIONS

CHORDS has been found to produce a more accurate representation of both high-resolution temperature and sound speed profiles than several other profile

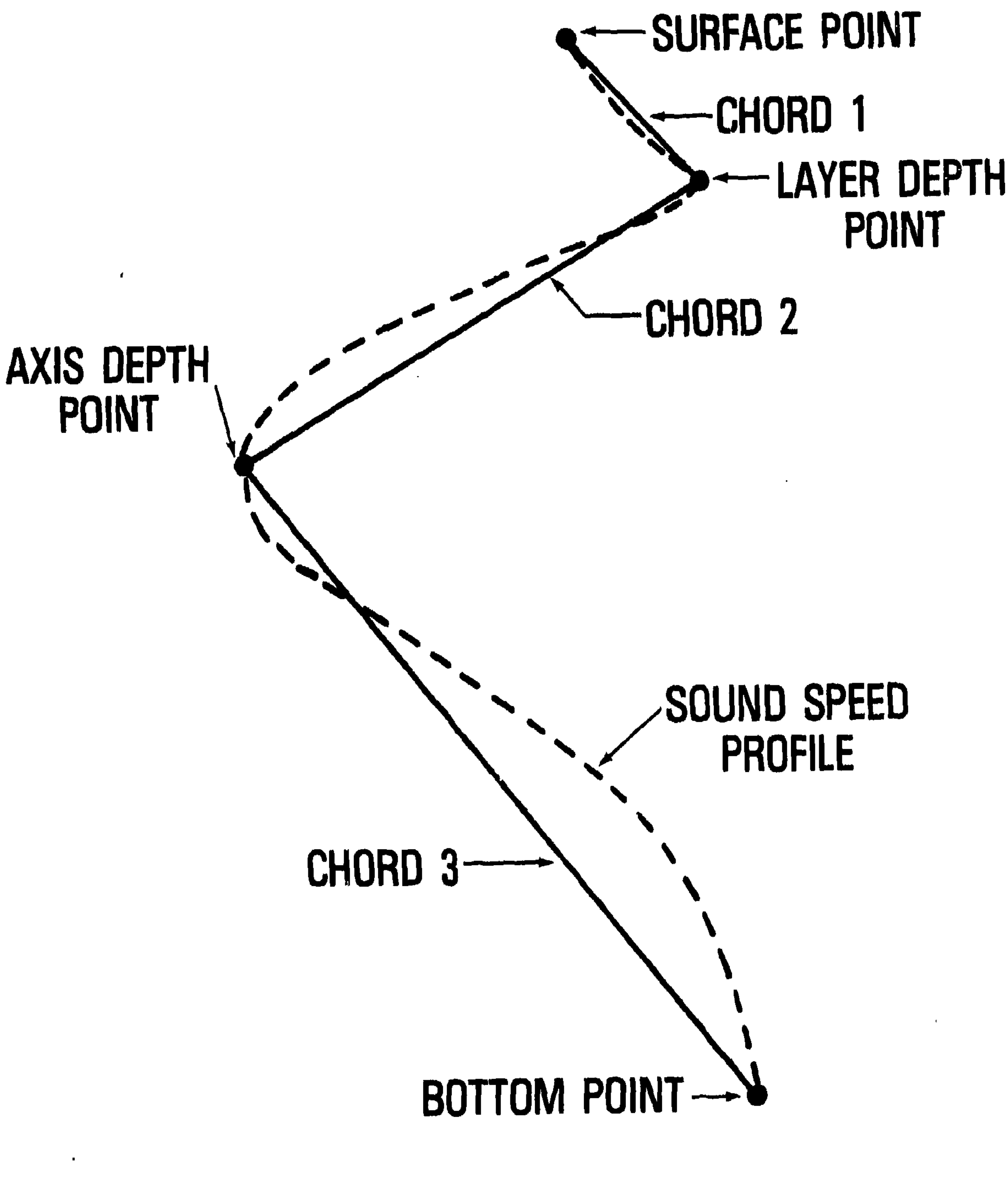

Figure 3. CHORDS initial significant point and chord sets

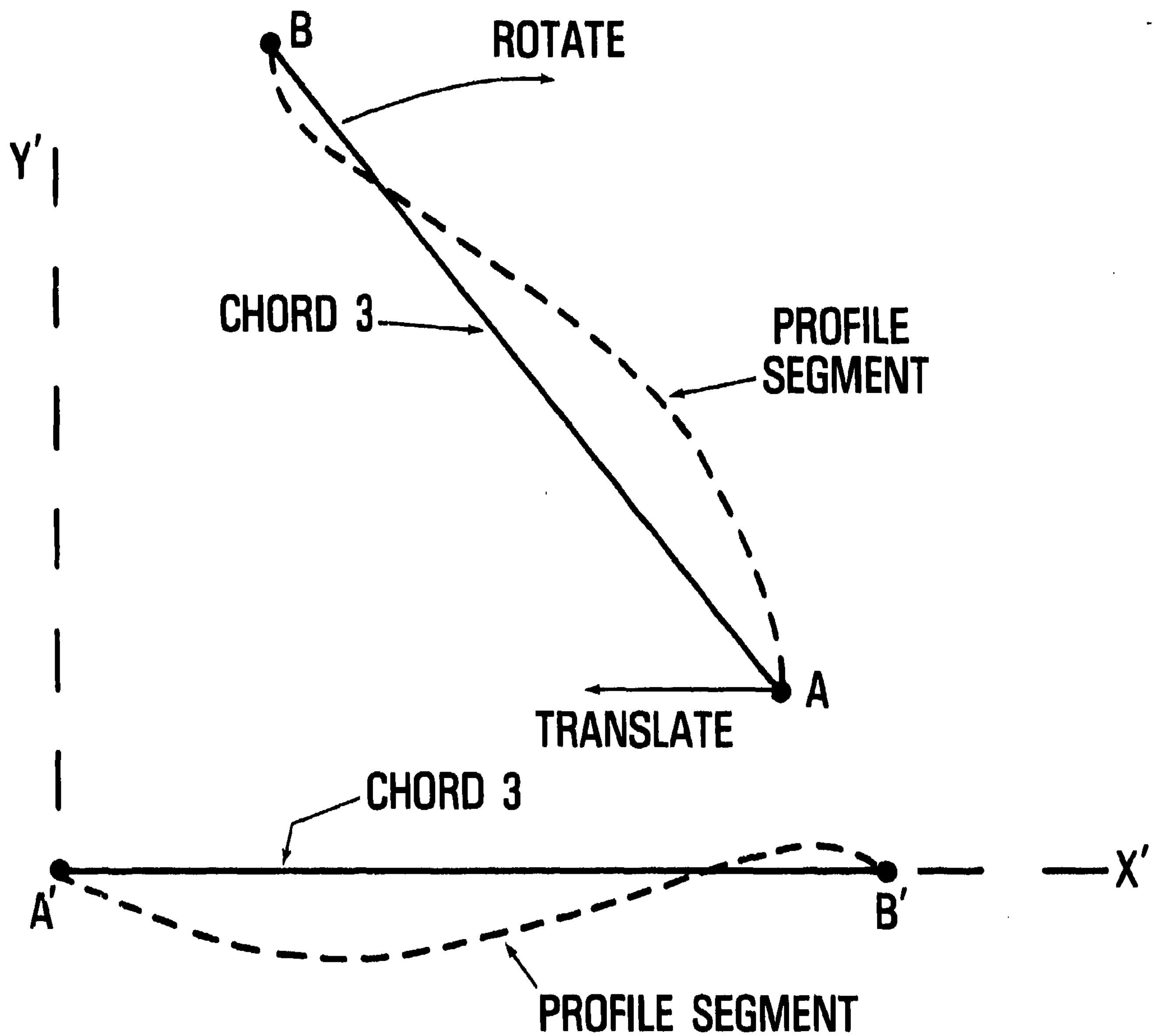

Figure 4. Translation and rotation of a chord and corresponding profile segment

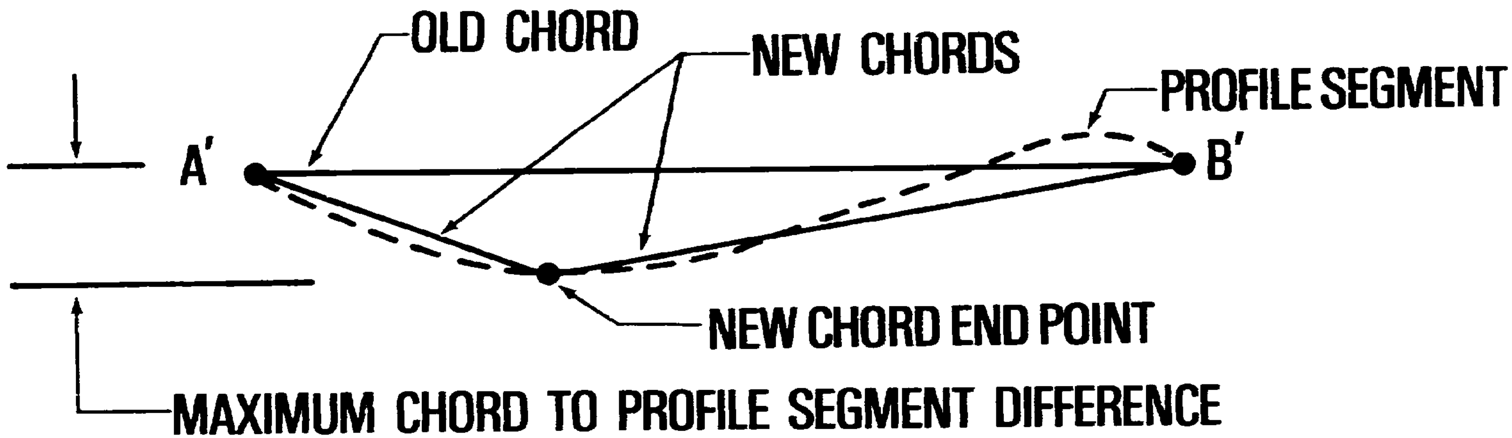

Figure 5. Selection of a new significant point leading to the replacement of an existing chord by two new chords

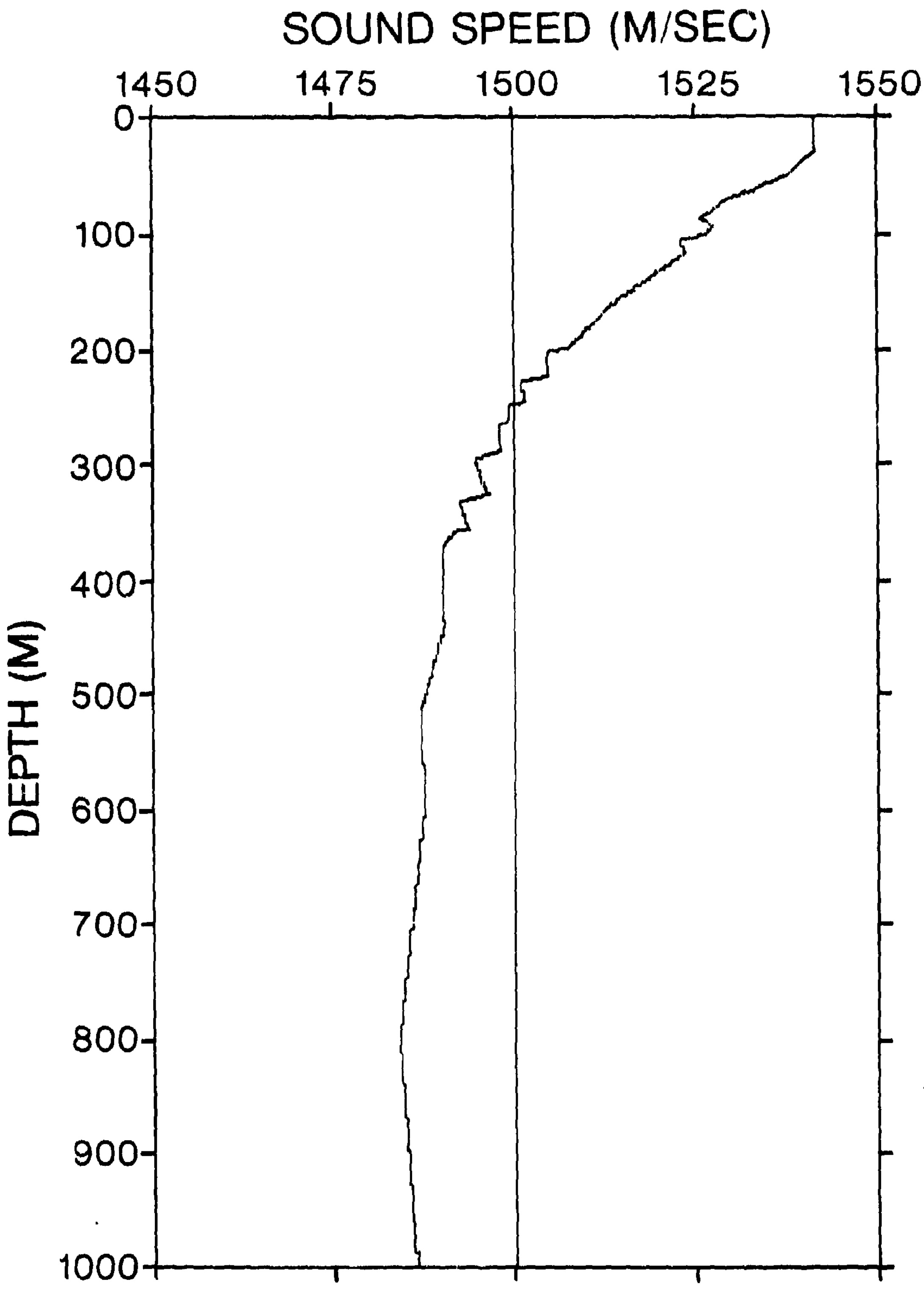

Figure 6. CHORDS thinned 30-point representation of the original profile (Fig. 1)

thinning algorithms in a limited number of test cases. Since profiles covering the extremes from fairly smooth to highly variable (the staircased profile presented in this report) were tested, it is reasonable to assume that CHORDS will also perform equally as well for profiles between these extremes. For low-frequency applications it is possible that other algorithms may be modified to enable them to better represent the deeper portions of the profiles being thinned. However, for high-frequency work or high-resolution highly variable input profiles, simple fixes may not be possible.

It is suggested that potential users of CHORDS pay particular attention to their own applications and test CHORDS accordingly. The tolerance values suggested and the number of output points required may not be optimal for all applications.

V. RECOMMENDATIONS

It is recommended that CHORDS be seriously considered in all applications (i.e., NORDAPS, TOPS input to SHARPS) where a reduction in the number of sound speed or temperature profile points is required.

The CHORDS concept may be easily applied to the thinning problems (i.e., density profiles, salinity profiles, high-resolution numerical model output, etc.). These other applications should be given additional thought and testing.

With the capability now available to capture and store high resolution (i.e., 1 meter temperature and sound speed) information, it is important to know the effect of thinning this data to some arbitrary lower resolution. It is therefore recommended that studies be made in areas where results of such thinning may have important consequences. One such area for study would be the relationship between transmission loss and the number of retained sound speed profile points. It is expected that such a study would result in the definition of the optimal number of profile points required to obtain accurate transmission loss results in various ocean areas. In addition to being area dependent, results are expected to depend on the closeness of the profile match at various profile depths as related to the effected modes of sound propagation.

APPENDIX A: CHORDS DETAILED DESCRIPTION

I. INPUT AND OUTPUT

CHORDS is a FORTRAN V callable subroutine written on a Digital Equipment Corporation VAX 11/780 minicomputer. The subroutine reduces the number of input sound speed profile points using the desired number of output points or a tolerance limit to cease processing. The tolerance limit refers to the maximum acceptable difference between a chord and its corresponding profile segment. By passing a positive tolerance limit to the subroutine, output profiles of varying numbers of points will result. The number of output points will depend on the tolerance limit selected and the complexity of the profile being thinned. Tolerance limit values of 0.1 for sound speed profiles and 0.05 for temperature profiles have reproduced test profile shapes sufficient for low frequency acoustic applications while reducing the number of profile points by a factor of 10. A zero value of the tolerance limit will force the subroutine to halt the thinning operation when the total number of desired output points has been reached.

There are three arguments in the subroutine call:

CALL CHORDS (NIN , NOUT, TOL), where

NIN = the user-supplied number of input profile points, NOUT = the number of output profile points determined by the algorithm, and
TOL = the user-supplied tolerance value.

The input profile is passed to the subroutine through the following named common area:

COMMON /DATA/ D(ASIZ),S(ASIZ), where

D is the input depth array, and
S is the input sound speed array.

The output profile is passed through the following named common area:

COMMON /DATAO/ DR(MNPTS),SR(MNPTS), where

DR is the output depth array, and
SR is the output sound speed array.

The program is set up to handle profiles consisting of a maximum of 1000 points. This number can be changed by modifying the parameter ASIZ, in all subroutines in which this parameter appears, to equal the maximum number of input profile points expected.

The maximum number of output profile points is presently set to 30. This number may be changed by modifying the parameter MNPTS appearing in the parameter statement in subroutine CHORDS. If enough profile points are not supplied to the subroutine, a zero value will be returned for the number of output points (NR).

II. DETAILED FUNCTIONAL DESCRIPTION

In the following discussion the numbers in parentheses refer to the program line numbers involved. A program listing can be found in Appendix B. A complete variable list and externals called can be found in Appendix C. A flow chart indicating the relationship between the subroutines and functions called can be found in Figure A1. The subroutine discussed is designed for use with sound speed profiles. The version designed for temperature profiles differs only in the selection of the initial set of chord points.

As outlined in Figure A1, CHORDS first chooses an initial set of chords. For sound speed profiles the initial set may be composed of from two to four points or from one to three chords.

The initial set of points will always contain the surface (depth = 0) point (93) and the last profile point (98). If the first profile point is not at the surface, then a surface value of sound speed is determined by a call to subroutine INTERP (67) which uses the first two profile points to extrapolate sound speed to the surface. The first profile point is replaced by the surface point. Other initial points could be the point defining the layer depth (84-88) and the point defining the deep sound channel axis depth (75-79). The deep sound channel axis depth is selected as that depth where the minimum sound speed is found. The layer depth is defined as that depth where the above axis maximum sound speed is found. These additional initial points will not be used if they coincide with the surface or last profile points.

The subroutine designed for use with temperature profiles differs from the above in that the only addition initial point which might be added to the surface and bottom points is a point defining the mixed layer depth. This depth is defined as that depth below which the temperature differs from the surface temperature by ±0.2°C.

For each initial chord, the encompassed profile segment point farthest away from the chord is located through a call to subroutine CHORD (144). Subroutine CHORD calls subroutine RTRAN which translates and rotates the profile segment points so that the chord becomes the new X axis. Subroutine RTRAN uses function ALPHA to determine the angle of rotation. The profile segment point lying farthest from the now horizontal chord can be easily found by locating the point of largest absolute magnitude. This magnitude and the corresponding profile point number are returned and stored in arrays DIFF and NUM respectively.

The resulting difference set (array DIFF) is then examined to determine which element possesses the largest chord to profile segment difference (188-191). The profile point at which this maximum difference is found (stored in array NUM) becomes an end point for two new chords (235-239). For one new chord, the point will be the first chord point. For the other new chord, the point will be the last chord point. The chord counter LINCNT is incremented by one (196) since the two new chords replace an existing chord, thereby increasing the total number of chords by one.

The final step for this first new chord, determined from the initial chord set, is to store the new chord end points (array PTPAIR), and make room in the difference set array (DIFF) and the point of maximum difference array (NUM) for the yet to be determined values of these variables for the two new chords. The section of code which handles the array insertion depends on where in the chord sequence the new chord is found (217-226). The section beginning at program line 227 and ending at program line 251 deals with the insertion of the two new chords in the arrays when the chord number counter J and the new chord sequence number (MAXLIN) match (221). The variable MAXLIN is used to determine in which original chord the new two chords are located. Variable LMAX is equated to MAXLIN and variable NMAX is made equal to MAXLIN plus one to retain the new chord sequence numbers for the next iteration. The program section beginning at line 256 and extending to line 268 bumps the existing arrays down after insertion of the new data. The program section beginning at line 272 and ending at line 280 inserts the last or bottom chord information in the arrays.

In the following iterations, the only chord to profile segment maximum differences that need to be calculated are for the two chords defined in the previous interation. These differences are found through the call to CHORD found at program line 177. The CHORD call is triggered by a match of the chord number counter J to either variable LMAX or NMAX (160). In the loop beginning at program line 149 the maximum chord to profile segment difference is again located for the chord set now including the two new chords (188-191). The chord-splitting process is repeated until either the total number of desired points have been located (206) or until the tolerance limit is reached (200).

Processing stops when either of the two above conditions are met. The resulting profile (arrays DR and SR), except for the last profile point, is generated from the first elements of the chord end point array (PTPAIR). The last profile point is defined by the last of the second elements of the same array.

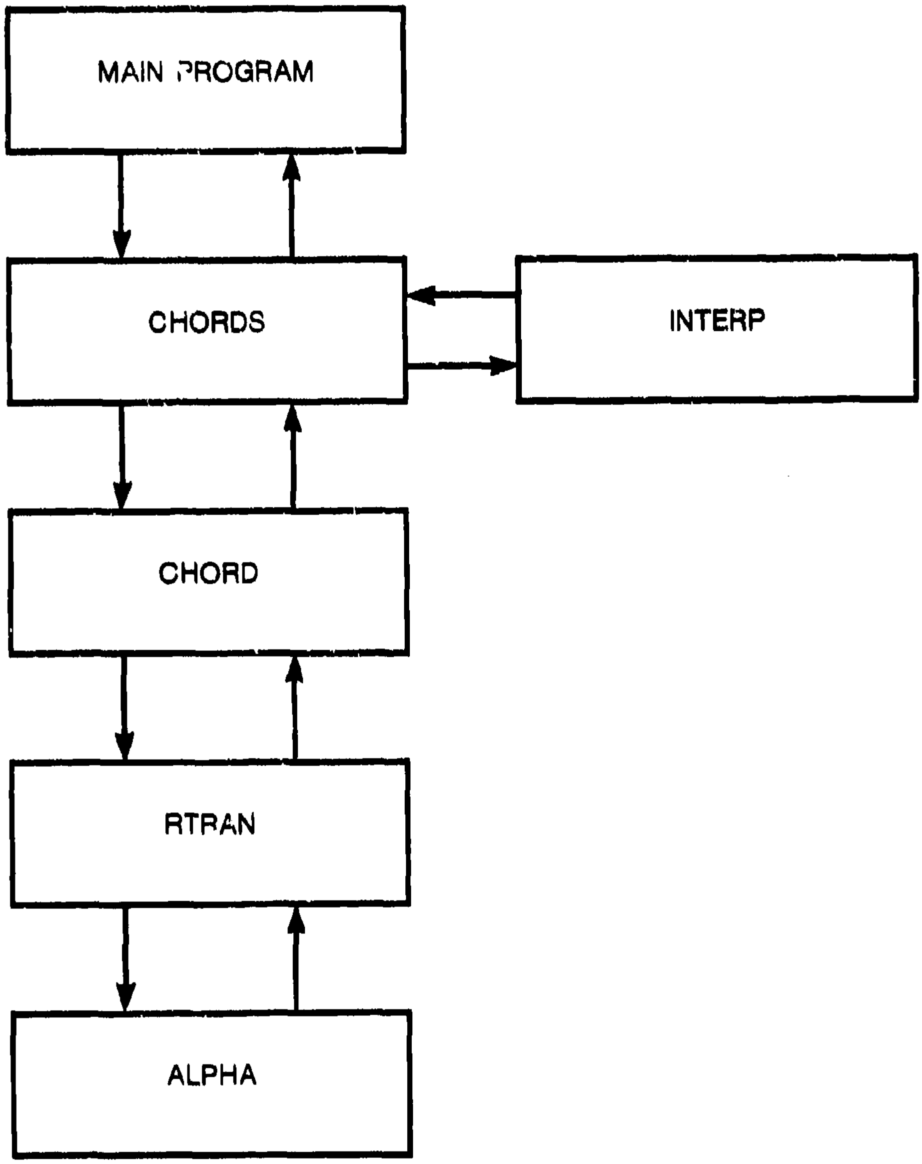

Figure A1. Functional relationship between the main calling program and CHORDS subroutines and functions

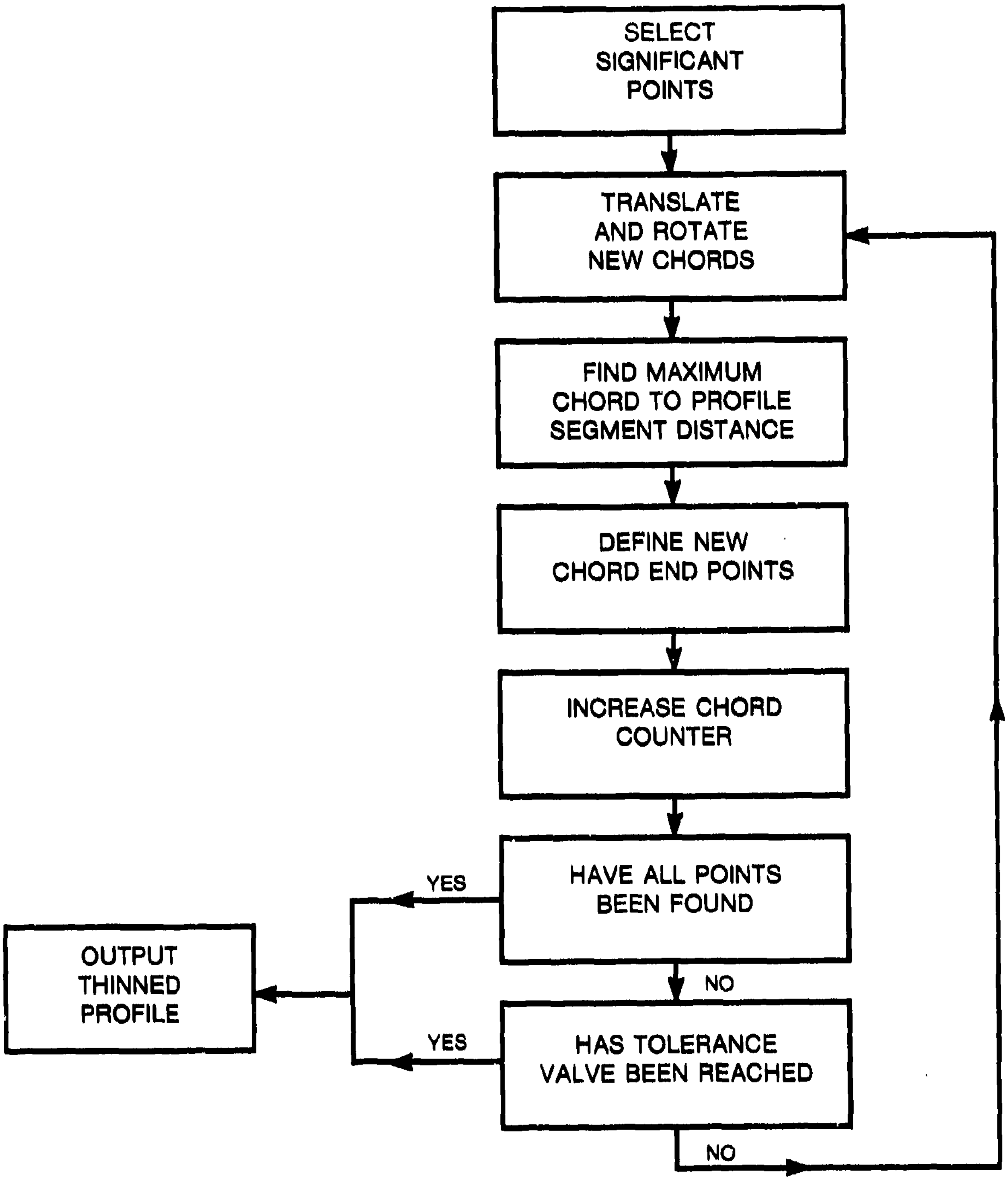

Figure A2. Logical flow diagram for the CHORDS point selection algorithm

APPENDIX B: SUBROUTINE LISTING

```
      SUBROUTINE CHORDS(NIN,NR,CTDLIM)
C
C           THIS SUBROUTINE PERFORMS A THINNING OF SMOOTHED
C           OR EDITED SOUND SPEED PROFILES. FOR A FILTERING
C           ALGORITHM USED FOR CTD 1 METER DATA CONTACT GEORGE KERR
C           NORDA CODE 323, NSTL, MS 39529, OR EUGENE
C           MOLINELLI, PSI, MCLEAN, VA.(AUTHOR OF THE INITIAL
C           FILTERING ALGORITHM).
C
C           THIS SUBROUTINE IS CURRENTLY SET UP TO THIN SOUND SPEED
C           PROFILES CONSISTING OF A MAXIMUM OF 1000 POINTS TO A
C           PROFILE CONSISTING OF EXACTLY 30 POINTS. THE MINIMUM
C           NUMBER OF INPUT PROFILE POINTS IS 32.
C
C           TO CHANGE THESE LIMITS PARAMETERS MPNTS(DEFINING THE NUMBER
C           OF POINTS TO FILTER TO), AND ASIZ(DEFINING THE MAXIMUM NUMBER
C           OF INPUT PROFILE POINTS), MUST BE CHANGED IN ALL SUBROUTINES
C           IN WHICH THESE PARAMETERS APPEAR.
C
C
C           THE MAIN PROGRAM CALLING THIS ROUTINE MUST CONTAIN
C           THE COMMON AREAS /DATA/, AND /DATAO/ TO HOLD THE RAW INPUT
C           DEPTHS (D) AND SOUND SPEEDS (S), AND THE OUTPUT DEPTHS (DR)
            AND
C           SOUND SPEEDS (SR) RESPECTIVELY.
C
C           THE ONLY ARGUMENTS PASSED TO THE SUBROUTINE ARE THE NUMBER
C           OF INPUT DATA POINTS (NIN) AND THE DESIRED TOLERANCE LIMIT
C           (CTDLIM). NR, RETURNED BY THE SUBROUTINE AS A CALLING
            ARGUMENT,
C           WAS INCLUDED TO RETURN THE NUMBER OF POINTS THINNED TO
C           IF THE TOLERANCE LIMIT RATHER THAN THE NUMBER OF POINTS TO
            THIN
C           TO WAS EMPLOYED TO HALT THE THINNING OPERATION.
C
C           SUBROUTINE CHORDS DEVISED AND PROGRAMMED BY GEORGE KERR
C           NORDA CODE 223, NSTL, MS 39529.(2MAY83)
C
      PARAMETER ASIZ=1000,MPNTS=30
C
      COMMON /DATA/ D(ASIZ),S(ASIZ)
      COMMON /DATAO/ DR(MPNTS),SR(MPNTS)
      COMMON /IPOINT/ IBEG,IEND
      INTEGER PTPAIR(MPNTS,2),TEMP1,TEMP2,TEMP3,TEMP4
      REAL INTERP
      DIMENSION DIFF(MPNTS),NUM(MPNTS)
      DATA DIFFLT/-99999./
      NOLIN=MPNTS-1
C
C           ZERO ARRAYS
C
      DO 9999 J=1,MPNTS
```

```
      DIFF(J)=0.
      NUM(J)=0
 9999 CONTINUE
      NR=0
C
C                DEFINE THE MINIMUM NUMBER OF POINTS ALLOWED
C
      MINPTS=NOLIN+3
C
C                CHECK TO SEE IF THE PROFILE HAS SUFFICIENT POINTS
C
      IF(NIN.LT.MINPTS) RETURN
C
C                IF THE PROFILE DOES NOT HAVE A SURFACE VALUE
C                EXTRAPOLATE TO FIND A SURFACE VALUE
C
      IF(D(1).EQ.0.0) GO TO 2500
      S(1)=INTERP(D(1),S(1),D(2),S(2),0.0)
 2500 D(1)=0.0
      SSMIN=99999.
      SSMAX=-99999.
C
C                FIND THE DEEP SOUND CHANNEL AXIS (DEFINED HERE AS
C                THE ABSOLUTE MINIMUM OF ALL SOUND SPEEDS INPUT)
C
      DO 3000 J=1,NIN
      IF(S(J).GE.SSMIN) GO TO 3000
      MINPT=J
      SSMIN=S(J)
 3000 CONTINUE
C
C                FIND THE LAYER DEPTH (DEFINED HERE AS THE ABSOLUTE
C                SOUND SPEED MAXIMUM ABOVE THE DEEP SOUND CHANNEL AXIS DEPTH)
C
      DO 3001 J=MINPT,1,-1
      IF(S(J).LE.SSMAX) GO TO 3001
      MAXPT=J
      SSMAX=S(J)
 3001 CONTINUE
C
C                RETAIN THE SURFACE POINT AS
C                THE FIRST POINT ON THE FIRST CHORD
C
      PTPAIR(1,1)=1
C
C                RETAIN THE BOTTOM POINT AS
C                THE LAST POINT ON THE FIRST CHORD
C
      PTPAIR(1,2)=NIN
C
C                INITIALIZE THE NUMBER OF CHORDS FOUND
C
      LINCNT=1
C
```

```
C              CHECK TO INSURE THAT THE LAYER DEPTH POINT IS
C              NOT THE SAME AS THE FIRST OR LAST PROFILE POINT
C
      IF(MAXPT.EQ.1.OR.MAXPT.EQ.NIN) GO TO 83
C
C              INSERT THE LAYER DEPTH DEFINED CHORDS IN THE CHORD
C              END POINT ARRAY
C
      PTPAIR(LINCNT,2)=MAXPT
      LINCNT=LINCNT+1
      PTPAIR(LINCNT,1)=MAXPT
      PTPAIR(LINCNT,2)=NIN
   83 CONTINUE
C
C              MAKE SURE THE DEEP SOUND CHANNEL AXIS DEPTH
C              POINT IS NOT THE SAME AS THE FIRST, LAST, OR
C              LAYER DEPTH POINT
C
      IF(MINPT.EQ.1.OR.MINPT.EQ.NIN.OR.MINPT.EQ.MAXPT) GO TO 80
C
C              INSERT THE DEEP SOUND CHANNEL AXIS DEFINED CHORDS
C              IN THE CHORD END POINT ARRAY
C
      PTPAIR(LINCNT,2)=MINPT
      LINCNT=LINCNT+1
      PTPAIR(LINCNT,1)=MINPT
      PTPAIR(LINCNT,2)=NIN
   80 CONTINUE
      DO 85 J=1,LINCNT
C
C              CHOOSE CHORD END POINTS SO THAT ALL POINTS
C              ALONG THE CHORD CAN BE IDENTIFIED.
C
      IBEG=PTPAIR(J,1)
      IEND=PTPAIR(J,2)
C
C              FOR EACH CHORD DETERMINE THE MAXIMUM DEVIATION FROM
C              THE ROTATED AXIS AND THE NUMBER OF THE POINT AT WHICH
C              THIS OCCURS.
C
      CALL CHORD(DIFF(J),NUM(J))
   85 CONTINUE
      NMAX=0
      LMAX=0
  110 DIFMAX=-99999.
      DO 200 J=1,LINCNT
C
C              LOOK FOR NEWLY DEFINED CHORDS FOR WHICH MAXIMUM
C              DIFFERENCES HAVE NOT BEEN FOUND. NEW CHORDS WILL
C              BE DEFINED BY A PREVIOUS COMPARISON OF ALL CHORD
C              DIFFERENCES. THE CURVE HAVING THE LARGEST DIFFERENCE
C              FROM ITS CHORD WILL DEFINE 2 NEW CHORDS. ONLY
C              THESE TWO NEW CHORDS NEED TO HAVE DIFFERENCES
C              CALCULATED. DIFFERENCES FOR THE REMAINING CHORDS
```

```
      C               HAVE ALREADY BEEN FOUND.
      C
              IF(J.EQ.NMAX.OR.J.EQ.LMAX) GO TO 154
              GO TO 155
      C
      C               CALCULATE NEW CHORD DIFFERENCES.
      C
       154    IBEG=PTPAIR(J,1)
              IEND=PTPAIR(J,2)
      C
      C               IF THE TWO END POINTS OF THE CHORD ARE ADJACENT
      C               NO FURTHER DIVISION OF THIS CHORD IS POSSIBLE.
      C               THEREFORE, DO NOT CALCULATE A DIFFERENCE.
      C
              IF((IEND-IBEG).EQ.1) GO TO 160
      C
      C               CALCULATE THE DIFFERENCE AND FIND THE POINT NUMBER FOR
      C               THE NEWLY FOUND CHORD.
      C
              CALL CHORD(DIFF(J),NUM(J))
              GO TO 155
      C
      C               FOR ADJACENT CHORD END POINTS
      C               SET THIS CHORD DIFFERENCE EQUAL TO THE DEFAULT VALUE
      C
       160    DIFF(J)=CTDLIM
      C
      C               LOOK FOR THE MAXIMUM DIFFERENCE AMONG ALL
      C               THE CHORD DIFFERENCES.
      C
       155    IF(DIFF(J).LT.DIFMAX) GO TO 200
              DIFMAX=DIFF(J)
              MAXLIN=J
              MAXPT=NUM(J)
       200    CONTINUE
      C
      C               INCREASE CHORD COUNTER
      C
              LINCNT=LINCNT+1
      C
      C               HAS THE TOLERANCE LIMIT BEEN REACHED
      C
              IF(DIFMAX.LE.CTDLIM) GO TO 500
      C
      C               INCREMENT THE CHORD COUNTER, AND CHECK TO SEE
      C               WHETHER THE MAXIMUM NUMBER OF CHORDS (MAX #PTS-1)
      C               HAS BEEN REACHED.
      C
              IF(LINCNT.GT.NOLIN) GO TO 500
      C
      C               NOTE THE CHORD NUMBER CONTAINING THE MAX. CHORD DIFF.
      C
              LMAX=MAXLIN
              NMAX=LMAX+1
```

```
      J=0
  300 J=J+1
C
C              HAVE ALL CHORDS BEEN CHECKED
C
      IF(J.GT.LINCNT) GO TO 110
C
C              DOES THIS CHORD CONTAIN THE MAX. DIFF.
C
      IF(J.EQ.MAXLIN) GO TO 120
C
C              HAS THE CHORD WITH THE MAXIMUM DIFF. BEEN FOUND
C
      IF(J.GT.MAXLIN) GO TO 130
      GO TO 300
  120 K=J+1
C
C              ARE WE ON THE LAST CHORD
C
      IF(K.EQ.LINCNT) GO TO 140
C
C              INSERT THE NEW POINT IN THE POINT PAIR ARRAY
C
      TEMP1=PTPAIR(K,1)
      TEMP2=PTPAIR(K,2)
      PTPAIR(J,2)=MAXPT
      PTPAIR(K,1)=MAXPT
      PTPAIR(K,2)=TEMP1
C
C              TEMP. STORE THE DIFFERENCE FOR THE POINT PAIR REPLACED
C              IN THE ARRAY
C
      DIFTMP=DIFF(K)
      DIFF(J)=DIFFLT
      DIFF(K)=DIFFLT
      NUMTMP=NUM(K)
      NUM(J)=0
      NUM(K)=0
      J=J+1
      GO TO 300
C
C              BUMP THE POINT PAIR, DIFF., AND MAX. POINT ARRAYS
C              DOWN.
C
  130 TEMP3=PTPAIR(J,1)
      TEMP4=PTPAIR(J,2)
      PTPAIR(J,1)=TEMP1
      PTPAIR(J,2)=TEMP2
      TEMP1=TEMP3
      TEMP2=TEMP4
      DIFTMP1=DIFF(J)
      DIFF(J)=DIFTMP
      DIFTMP=DIFTMP1
      NUMTM1=NUM(J)
```

```
      NUM(J)=NUMTMP
      NUMTMP=NUMTM1
      GO TO 300
C
C              TAKE CARE OF THE LAST POINTS IN THE ARRAYS
C
 140  TEMP2=PTPAIR(J,2)
      PTPAIR(J,2)=MAXPT
      PTPAIR(K,1)=MAXPT
      PTPAIR(K,2)=TEMP2
      DIFF(J)=DIFFLT
      DIFF(K)=DIFFLT
      NUM(J)=0
      NUM(K)=0
      GO TO 110
 500  NR=LINCNT
C
C              USING THE POINT PAIR ARRAY, EXTRACT THE NEW
C              POINTS THAT REPRESENT THE THINNED PROFILE.
C
      L=LINCNT-1
      DO 2222 J=1,L
      DR(J)=D(PTPAIR(J,1))
      SR(J)=S(PTPAIR(J,1))
 2222 CONTINUE
      DR(NR)=D(PTPAIR(L,2))
      SR(NR)=S(PTPAIR(L,2))
      RETURN
      END
```

```
      REAL FUNCTION INTERP (P1,Q1,P2,Q2,P3)
C
C     (P1,Q1) AND (P2,Q2) ARE PAIRS BETWEEN WHICH OR PAST WHICH THE
C     INTER/EXTRA -POLATION IS DONE. P3 IS THE INTER/EXTRA -POLATED
C     POSITION.
C
C
      IF(P1.NE.P2) GO TO 10
      INTERP=Q1
      RETURN
 10   CONTINUE
      FACT1=P3-P1
      FACT2=P2-P1
      FACT=FACT1/FACT2
      ANS1=Q2-Q1
      INTERP=FACT*ANS1+Q1
C
      RETURN
      END
```

```
      SUBROUTINE CHORD(YMAX,NUM)
      PARAMETER ASIZ=1000
      COMMON /DATA/ Y(ASIZ),X(ASIZ)
      COMMON /PRIM/ XPRIM(ASIZ),YPRIM(ASIZ)
      COMMON /IPOINT/ IBEG,IEND
C
C           DO AXIS TRANSLATION AND ROTATION
C
      CALL RTRAN
C
C           FIND THE MAXIMUM DEVIATION OF THE TRANSFORMED
C           PROFILE SEGMENT FROM THE TRANSFORMED CHORD
C
      YMAX=-9999.
      DO 10 J=IBEG,IEND
      IF(ABS(YPRIM(J)).LE.YMAX) GO TO 10
      YMAX=ABS(YPRIM(J))
C
C           NOTE THE POINT NUMBER OF THE MAXIMUM DEVIATION
C           POINT
C
      NUM=J
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE RTRAN
C
C           THIS SUBROUTINE TRANSLATES AND ROTATES THE LINE
C           SEGMENTS SO THAT THE CHORD BECOMES THE NEW X
C           AXIS. THE LAST POINT ON THE LINE WILL
C           COINCIDE WITH THE NEW AXIS ZERO (X'=0,Y'=0)
C
      PARAMETER ASIZ=1000
      COMMON /DATA/ Y(ASIZ),X(ASIZ)
      COMMON /PRIM/ XPRIM(ASIZ),YPRIM(ASIZ)
      COMMON /IPOINT/ IBEG,IEND
      DO 10 J=IBEG,IEND
C
C           TRANSLATE THE POINTS SO THAT THE LAST POINT ON THE
C           LINE COINCIDES WITH 0,0 ON THE NEW AXIS
C
      XTEMP=X(J)-X(IEND)
      YTEMP=Y(IEND)-Y(J)
C
C           IF THIS IS THE FIRST POINT OF THE LINE CALCULATE
C           THE ANGLE OF ROTATION REQUIRED TO BRING THE CHORD
C           DOWN TO THE NEW X AXIS.
C
C           FOR THE FIRST POINT XTEMP IS DELTA X ALONG THE LINE
C           AND YTEMP IS DELTA Y ALONG THE LINE, WHERE THE SLOPE
```

```
C               OF THE LINE BETWEEN THE TWO END POINTS IS
C               (DELTA Y) / (DELTA X).
C
      IF(J.EQ.IBEG) A=ALPHA(XTEMP,YTEMP,0.,0.)
C
C               FIND THE NEW ROTATED COORDINATES FOR EACH POINT
C
      XPRIM(J)=XTEMP*COS(A)+YTEMP*SIN(A)
      YPRIM(J)=-XTEMP*SIN(A)+YTEMP*COS(A)
C
C               IF THE LINE SEGMENT MUST BE ROTATED BY 90 DEGREES
C               INSURE THAT THE NEW Y COORDINATE IS ZERO AT THE
C               END POINTS.
C
      IF(J.EQ.IBEG.AND.A.EQ.90.) YPRIM(J)=0.
      IF(J.EQ.IEND.AND.A.EQ.90.) YPRIM(J)=0.
   10 CONTINUE
      RETURN
      END
```

```
      FUNCTION ALPHA(X1,Y1,X2,Y2)
C
C               THIS FUNCTION CALCULATES THE ANGLE OF ROTATION
C               REQUIRED TO ROTATE THE CHORD SO THAT IT LIES
C               ON THE X AXIS.
C               INPUT ARE THE X AND Y COORDINATES OF THE LEFT
C               MOST POINT (X1,Y1) AND THE RIGHT MOST POINT
C               (X2,Y2).
C
C               CHECK TO INSURE AN INFINITE RESULT DOES NOT OCCUR
C
      IF(X1.EQ.X2) GO TO 1
      ALPHA=ATAN((Y1-Y2)/(X1-X2))
      RETURN
    1 ALPHA=90.
      RETURN
      END
```

APPENDIX C: VARIABLE AND EXTERNAL CALL DESCRIPTION

I. CHORDS VARIABLES

Except as noted, real variables begin with the letters A through H, and O through Z. Integer variables begin with the letters I through N.

CTDLIM - The tolerance value used to halt processing

DIFFLT--A default value used to fill unused portions of the maximum difference array (DIFF)

DIFMAX--The maximum value of all chord to profile segment differences

DIFTMP and DIFTMP1--Temporary storage locations for the chord to profile segment differences stored in array DIFF

IBEG--Beginning point number of a chord delineated profile segment

IEND--Ending point number of a chord delineated profile segment

J,K, and L--DO loop variables or integer counters

LINCNT--A counter used to keep track of the number of chords found

LINLIM--A constant equal to the total number of chords allowed

LMAX--Used to store the chord number containing the maximum chord to profile segment difference of all such sets

MAXLIN--Same as LMAX above

MAXPT--The input profile point number defining the layer depth

MINPT--The input profile number point defining the deep sound channel axis depth

NMAX--Used to store the chord number one greater than that having the maximum chord to profile segment difference

NR--The number of points in the output thinned profile

NUMTM1 and NUMTMP--Temporary storage locations for the number of the chord containing the maximum chord to profile segment difference

SSMAX--Temporary storage location for the value of the input profile's maximum above axis sound speed

SSMIN--Temporary storage location for the value of the minimum input profile sound speed

TEMP3, TEMP4, TEMP1, and TEMP2--Integer temporary storage locations for the beginning and ending chord points

II. CHORDS ARRAYS

D--Input profile depth array

DIFF--Used to store the chord to profile segment maximum differences

DR--Output profile depth array

NUM--Used to store the point numbers of the maximum chord to profile segment differences

PTPAIR(N,2)--The array containing the beginning chord point numbers in elements (N,1), and the ending chord point numbers in elements (N,2)

S--The input sound speed array

SR--The output sound speed array

III. CHORDS REFERENCED SUBROUTINES AND FUNCTIONS

Function INTERP(X1,Y1,X2,Y2,X3)

Interpolates (or extrapolates) between two given points (X1,Y1) and (X2,Y2) to find the Y value at X3. Used to determine the sound speed at the surface when it is not given.

Subroutine CHORD(YMAX,NUM)

This subroutine locates the maximum absolute difference between a chord and its corresponding profile segment.

Variables

IBEG--Beginning chord profile point number

IEND--Ending chord profile point number

NUM--The number of the profile point having the maximum absolute difference from its chord

YMAX--Maximum absoulute difference between a chord and its corresponding profile segment

Arrays

X--Profile segment depth values

XPRIM--Transformed (translated and rotated) profile segment depth values

Y--Profile segment sound speed values

YPRIM--Transformed (translated and rotated) profile segment sound speed values

Subroutines and functions referenced

RTRAN--(see below)

Subroutine RTRAN

Transforms the sound speed profile segment points so that the corresponding chord becomes the new X axis

Variables

A--Angle of rotation of the chord and profile segment such that the translated chord becomes the new horizontal (X) axis

IBEG--Beginning profile segment point

IEND--Ending profile segment point

J--Loop counter

XTEMP--Temporary storage location of the profile segment's transformed sound speed values

YTEMP--Temporary storage location of the profile segment's transformed depth values

Arrays

X--Profile segment sound speed values

XPRIM--Transformed (translated and rotated) profile segment sound speed values

Y--Profile segment depth values

YPRIM--Transformed (translated and rotated) profile segment depth values

Subroutines and functions referenced

Function ALPHA

This function calculates the angle of rotation required to rotate the chord so that it becomes parallel to the X axis